Stosy i kolejki. Laboratorium 07

Dr hab. Bożena Woźna-Szcześniak,prof UJD

May 18, 2023

1. Napisz program testujący klasę generyczną `Stack`, tak aby dla wejścia podanego jako ciąg znaków:

   ```
   Nie zda sie na nic wypracownie - - -
   Gdy z liter - - nawet slad nie - zostanie
   ```

   na wyjściu pojawił się następujący wynik:

   ```
   wypracownie nic na liter z nie (7 pozostalo na stosie)
   ```

   ```
   import java.util.Scanner;
   public class Dash {
     public static void main(String[] args) {
      ...
     }
   }
   ```

2. Napisz program `Nawiasy.java` wykorzystującą klasę generyczną `Stack` DLA ZNAKÓW (tj. `<Character>`), która odczytuje sekwencję prawych i lewych nawiasów okrągłych (tj. (,)), nawiasów klamrowych (tj. {, }) i nawiasów kwadratowych (tj. [,]) ze standardowego wejścia i używa stosu, aby sprawdzić, czy sekwencja jest prawidłowo zrównoważona. Na przykład: Dla ciągu `[()] {} {[() ()] ()}` program powinien wypisać wartość true , a dla ciągu `[(])` wartość false.

   ```
   public class Nawiasy {
       private static final char LEWY_OKR     = '(';
       private static final char PRAWY_OKR    = ')';
       private static final char LEWY_KLAMR     = '{';
   ```

```
    private static final char PRAWY_KLAMR    = '}';
    private static final char LEWY_KWADR   = '[';
    private static final char PRAWY_KWADR  = ']';

    public static boolean isBalanced(String s) {
        ...
    }
    public static void main(String[] args) {
        ...
    }
}
```

3. Zmień i rozszerz program `Dijkstra2StackAlgorithm` z wykładu tak aby, korzystał z generycznej wersji klasy `Stack` i obliczał w pełni nawiasowane wyrażenie arytmetyczne zawierające dodatkowo operację liczenia procentu, operację dzielenia modulo, obliczanie logarytmu naturalnego, obliczanie logarytmu przy podstawie 2, obliczanie logarytmu przy podstawie 10.

4. Zmień i rozszerz program `InfixToPostfix` z wykładu tak aby, korzystał z generycznej wersji klasy `Stack` i obliczał postać postfiksową dla formuły klasycznego rachunku zdań. Przykład: ((p or q) and s) powinno zostać zastąpione p q or s and. ((1 or 0) and 1) powinno zostać zastąpione 1 0 or 1 and. **Uwaga !** Uwaga na wykładzie czytaliśmy wyrażenie zakończone znakiem 'q' !

5. Zmień i rozszerz program `EaluatePostfix` z wykładu tak, aby obliczał wartość wyrażeń Boolowskich. Np., wartością wyrażenia 1 0 or 1 and jest true; wartością wyrażenia 1 0 and 1 or jest true; wartością wyrażenia 1 0 and 0 or jest false.

6. Napisz program `Ogon.java`, aby `Ogon k <file.txt` wypisał ostatnie k wierszy pliku `file.txt`.

7. Zaimplementuj kolejkę przy pomocy dwóch stosów tak, aby każda operacja kolejkowa przyjmowała stałą amortyzowaną liczbę operacji na stosie.